The ArcIMS 3 Architecture

An ESRI White Paper • May 2000

ESRI 380 New York St., Redlands, CA 92373-8100, USA • TEL 909-793-2853 • FAX 909-793-5953 • E-MAIL info@esri.com • WEB www.esri.com

U.S. GOVERNMENT RESTRICTED/LIMITED RIGHTS

J-8488

The ArcIMS 3 Architecture

An ESRI White Paper

Contents **Page**

Contents **Page**

The ArcIMS 3 Architecture

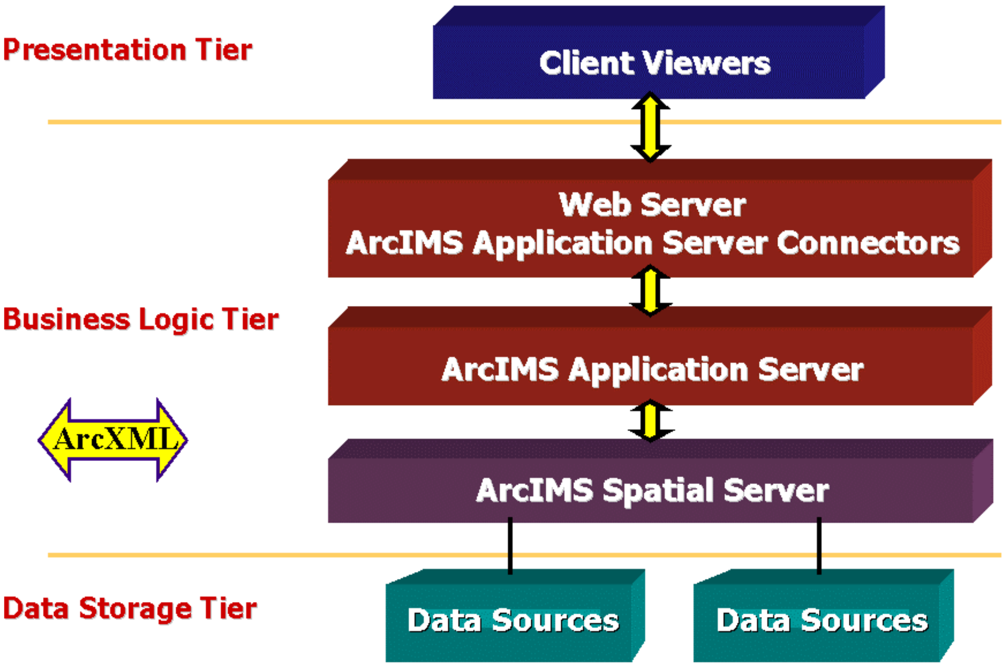
Introduction

The ArcIMS™ architecture has been specifically engineered to serve geographic information system (GIS) data and services on the Internet. As ESRI's first stand-alone GIS and mapping solution, a description of how the different components interact with one another will assist in understanding the system. This white paper will address how to best configure and scale ArcIMS sites, how to distribute ArcIMS components, and how ArcIMS fits into an existing Internet configuration.

Multitier Architecture

ArcIMS has a multitier architecture consisting of presentation, business logic, and data storage tiers. The presentation tier includes the ArcIMS Viewers. The business logic tier includes the Web Server, ArcIMS Application Server, and ArcIMS Application Server Connectors. The data storage tier includes the ArcIMS Spatial Server and any data sources. Communication between the tiers is handled through ArcXML, which is the ArcIMS version of XML or eXtensible Markup Language.

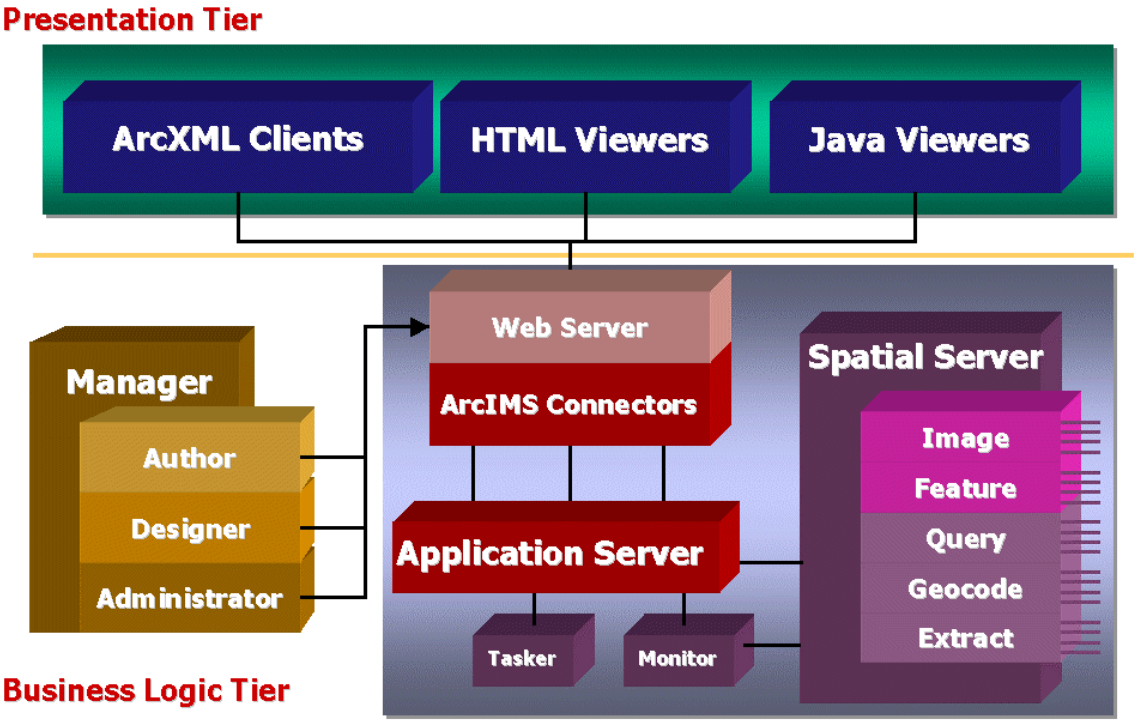
Multitier System



ArcIMS Components: Overview

ArcIMS operates in a distributed environment and consists of both client and server components. The client consists of the presentation tier components. The ArcIMS Viewers are used to send requests and to view maps and data. The three viewer types are ArcXML clients, HTML/DHTML Viewers, and Java Viewers including ArcExplorer™ 3 software.

ArcIMS Components



The serverside consists of components in the business logic tier: the ArcIMS Application Server Connectors, Application Server, Spatial Server, and Manager as well as a Web Server. The server components are used to process requests, create and run MapServices, and manage the site.

Programming Languages

ArcIMS components are written using a combination of Java 2 and C++. The following lists the development language used for each component.

Programming Languages Used for ArcIMS Components

Component	***Language***
Spatial Server	C++
Application Server	Java 2
Servlet Connector	Java 2
ColdFusion Connector	C++
ActiveX Connector	C++
Java Applet in Viewers	Java 2
Author, Designer, Administrator	Java 2
Manager	HTML
Viewers	JavaScript, HTML, DHTML

Using ArcXML with ArcIMS

ArcIMS uses ArcXML to communicate with the different components. ArcXML files appear similar to HTML pages; however, the difference is that HTML describes the page structure for display, while ArcXML provides the structure for describing the content.

ArcXML tags and attributes provide the structure for the following:

- **MapService configuration files** describe how a map should be rendered, which includes a list of layers used and how each layer should be symbolized.

- **Requests** set a filter on an existing MapService configuration file that specifies which part of a map and associated data will be acted upon.

- **Responses** send the information back to the client.

The following is a sample of ArcXML.

```
<ARCXML version="1.0">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
   <ENVELOPE minx="-180" miny="-90" maxx="180" maxy="90" />
      </PROPERTIES>
   </GET_IMAGE>
  </REQUEST>
</ARCXML>
```

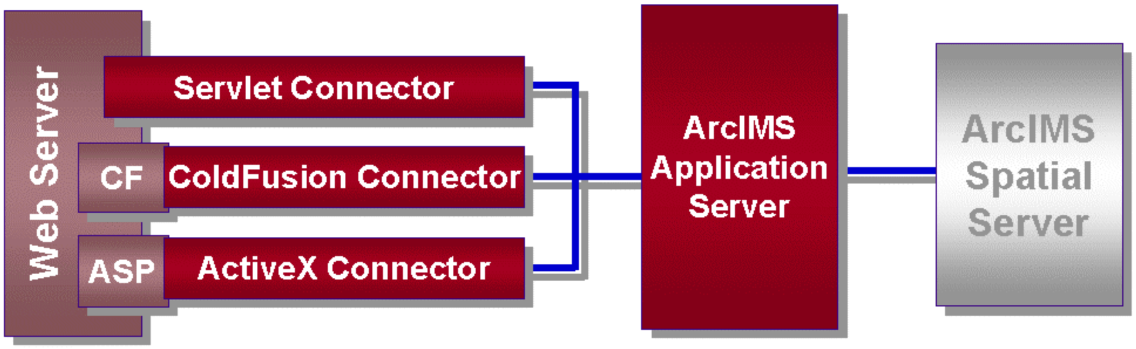
ArcIMS Components in the Business Logic Tier

The ArcIMS business logic tier contains the components that are needed to process requests and responses and to run MapServices. These components include the ArcIMS Application Server, the Application Server Connectors, and the Spatial Server. Although not included with ArcIMS, a Web Server and Java servlet engine are an integral part of this tier.

The ArcIMS Business Logic Tier (see diagram) shows the relationship between the different components. When an ArcIMS request is made, the request is first handled by the Web Server, passed through one of the connectors, and then handed to the ArcIMS

Application Server. The Application Server hands the request to an ArcIMS Spatial Server. The Spatial Server will be discussed in the ArcIMS Spatial Servers and Virtual Servers section.

ArcIMS Business Logic Tier



Web Servers

A Web Server must be able to communicate with one of the ArcIMS Application Server Connectors. With a typical ArcIMS installation, the Web Server needs to support Java servlet engine or have a native Java servlet. If only the ColdFusion or the ActiveX Connectors are used, a Java servlet is not needed, but the Web Server must be able to communicate with ColdFusion or Active Server Pager (ASP).

Supported Web Servers and Java Servlet Engines

Web Server	*Platforms*	*Servlets*
Microsoft IIS	Windows NT Server	JRUN, ServletExec
Netscape Enterprise Server and iPlanet	Windows NT Server and Workstation, Sun Solaris	Native Servlet
Apache	Windows NT Server and Workstation, Sun Solaris	JServ
Note: *IBM WebSphere is also supported and can be used in place of the Java servlet engines. WebSphere runs on IIS, Netscape Enterprise, and Apache.*		

ArcIMS Application Server

The Application Server runs as a background process (Windows NT service/UNIX daemon) and handles the load distribution of incoming requests. It also serves as a catalog for keeping track of which MapServices are running on which ArcIMS Spatial Servers. Using this information, the Application Server will hand off an incoming request to the appropriate Spatial Server.

Multiple Web Servers can communicate with the Application Server, and in keeping with the distributed environment philosophy, the Application Server can reside on a different computer from the Web Servers.

ArcIMS Application Server Connectors

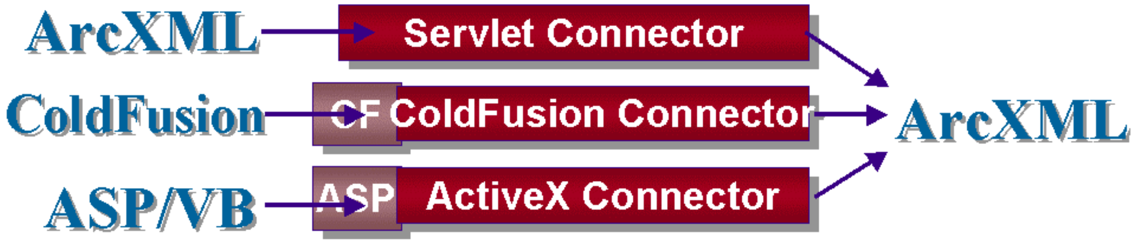
The connectors provide a communication pathway between a Web Server or a third party application server and the ArcIMS Application Server.

ArcIMS has three connectors:

- Servlet
- ColdFusion
- ActiveX

Each request establishes a socket connection with the ArcIMS Application Server. Once the channel is established, requests are sent to and responses are received from the Application Server. All connectors must reside on the same machine as a Web Server.

The Application Server expects requests to be written in ArcXML. Each connector is responsible for translating a third party language, such as ColdFusion or ASP, into ArcXML before handing the request to the Application Server.



ArcIMS connectors translate a third party language to ArcXML before the request is handed to the ArcIMS Application Server.

Servlet Connector

The Servlet Connector uses a Java servlet engine as the communication link between the Web Server and the ArcIMS Application Server. Servlet engines are a Java platform technology for extending Web Servers, and ArcIMS communicates with Web Servers through these servlet extensions. While each brand of servlet engine may have unique features, they all use the same application programming interface (API). Because servlets use the same protocol, ArcIMS can communicate easily with just about any Web Server that can communicate with a Java servlet engine.

The Servlet Connector is the default connector for ArcIMS. Requests and responses are written in ArcXML before reaching the connector and do not require additional translation. (The Servlet Connector is a Windows NT and UNIX solution.)

ColdFusion Connector

The ColdFusion Connector translates ColdFusion requests into ArcXML and passes the requests to the ArcIMS Application Server. Once a request is made, the Web Server hands the request to the ColdFusion Application Server for processing. When the ColdFusion Application Server spots an ArcIMS custom tag, the information is passed to the ColdFusion Connector and translated to ArcXML before being handed to the Application Server.

After the request is processed, a response returns (through the same channels) to the ColdFusion Application Server. When the ColdFusion Application Server receives the response, an HTML page is generated and sent to the Web Server. (The ColdFusion Connector is both a Windows NT and UNIX solution.)

ActiveX Connector

The ActiveX Connector is a Component Object Model (COM) Dynamic Link Library for use with COM applications such as Microsoft ASP. Using the ActiveX Connector, a connector object can be created in ASP, Visual Basic, C++, Delphi, or another COM-compliant language. Once in the development environment, a connector object is created to establish a connection with the Application Server. Then requests can be made to handle map operations using the ActiveX connector object model API. The ActiveX Connector translates these requests to ArcXML before handing them to the Application Server. When the application receives the response, an HTML is generated and sent onward to the Web Server. (The ActiveX Connector is available on Windows NT only.)

Additional Processes on the Server

In addition to the Application Server, two more background processes (Windows NT services/UNIX daemons) are used for supporting the Spatial Server.

- **ArcIMS Monitor** is used to track the state of the ArcIMS Spatial Server. The purpose of Monitor is to start new MapServices and Spatial Servers. When the system is rebooted, the MapServices are restored automatically through the ArcIMS Monitor.

- **ArcIMS Tasker** is used to removes output image files. These files, generated by the Spatial Server to support Image MapServices, are removed at a user-defined time interval.

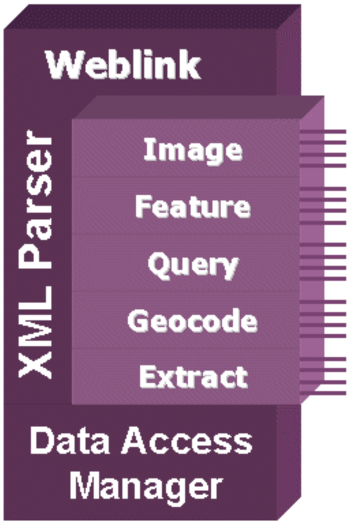
ArcIMS Spatial Servers and Virtual Servers

The backbone of ArcIMS is the ArcIMS Spatial Server. The Spatial Server provides the functional capabilities for accessing and bundling maps and data into the appropriate format before sending the data to a Web browser.

The Spatial Server can be distributed across an ArcIMS site in several ways. The simplest scenario is to have one Spatial Server on a machine. There can be multiple instances of the same Spatial Server running on the same machine. In a second scenario, more than one Spatial Server can run on the same machine. Again, there can be multiple instances of each Spatial Server running. A third scenario is to have one or more Spatial Servers and instances running on multiple machines.

The Spatial Server is a container for holding several different comp onents. Supporting functions include Weblink, the XML parser, and the Data Access Manager. Weblink is the communication gateway between the ArcIMS Application Server and the Spatial Server. The XML parser is used for parsing ArcXML requests. The Data Access Manager provides a link between the Spatial Server and any data sources.

ArcIMS Spatial Server



When a request is received, an ArcIMS Server may perform several functions.

- **Image Rendering** generates and sends maps to Web browsers as JPEG, PNG, or GIF images. Cartographic images can be generated from shapefiles, ArcSDE™ data sets, and supported image formats (e.g., ADRG, ASRP, BIL, BIP, BMP, BSQ, CADRG, CIB, ERDAS GIS and LAN, ERDAS IMAGINE, GeoTIFF, GIF, IMPELL, JFIF (JPEG), MrSID, NITF, Sun, TIFF, TIFF 6.0, USRP, and GRID).

- **Feature Streaming** sends shapefiles and ArcSDE data sets in a compressed format to a Java Applet in the client Web browser. Feature streaming is a temporary compressed format that remains only as long as the Java Applet is open. The Java Applet receives instructions on how to assemble the data once it is received. Feature streaming allows for more functional capabilities on the client such as clientside labeling, changing the appearance of a map, Map Tips, and clientside spatial selection.

- **Geocoding** locates addresses on maps. The geocoding function provides address, intersection, city, state, and place georeferencing services based on address information in shapefiles and ArcSDE data sets. The Geocode Server returns either an exact match or a list of candidate matches based on user inputs.

- **Query** returns associated data for spatial and tabular queries. Queries can be built against shapefiles, ArcSDE data sets, and joined external tables. This query function is required for handling attributes when image rendering is used.

- **Data Extraction** returns data in shapefile format. A request is sent to the server to extract data from shapefiles and ArcSDE layers, and the requested data is sent back to the client. This process is different from feature streaming because data is sent to the client as a zipped shapefile.

MapServices

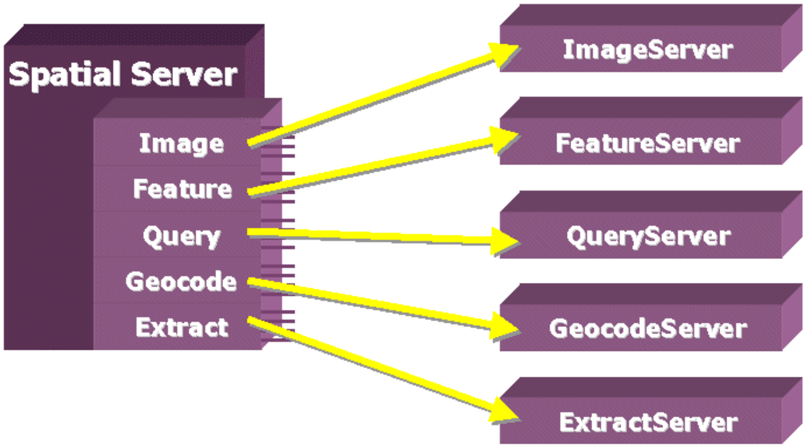
A MapService is a process that runs on the ArcIMS Spatial Server. It provides instructions to a Spatial Server on how to draw a map when a request is received. The input to a MapService is an ArcXML MapService Configuration File. While the ArcXML file and MapService are closely related, they are independent because the same ArcXML file can be used as input to more than one MapService.

ArcIMS supports two types of MapServices: Image and Feature. An Image MapService uses the image rendering capabilities of the Spatial Server. When a request is received, a map is generated on the server, and the response is returned as a JPEG, PNG, or GIF image. A new map image is generated each time a client requests more information.

Rather than rendering a map on the server, a Feature MapService uses the Spatial Server's feature streaming capabilities to bundle data and send the request to the client. Because more processing is performed in the Java Applet, requests are sent to an ArcIMS Spatial Server only when additional data is needed.

ArcIMS Virtual Servers

An ArcIMS Virtual Server is a grouping of one or more Spatial Servers. The different components of an ArcIMS Spatial Server are assigned to Virtual Servers for administration and to help with load distribution. Although not a physical entity, a Virtual Server is used to make it easier to manage Spatial Servers and MapServices. Because multiple Spatial Servers can be run on multiple machines, a mechanism such as a Virtual Server is needed to manage MapServices in this environment. When ArcIMS is installed, the software creates five Virtual Servers: ImageServer, FeatureServer, QueryServer, GeocodeServer, and ExtractServer.



ArcIMS installs five Virtual Servers.

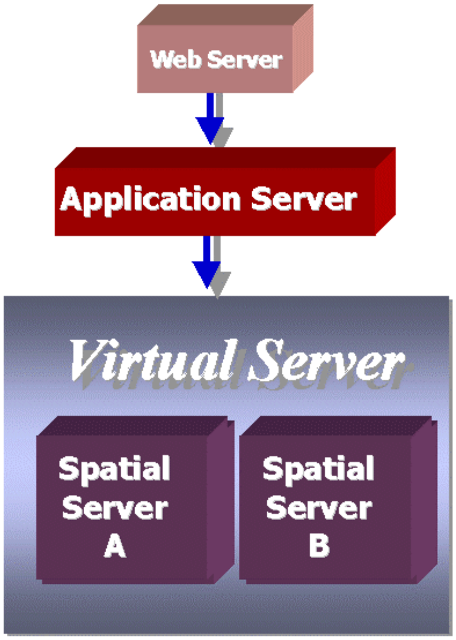
Additional Virtual Servers can be created as needed to help with MapService and Spatial Server administration.

A MapService is assigned to either an ImageServer or a FeatureServer rather than directly to an individual ArcIMS Spatial Server. The MapService will start on all the Spatial Servers within the Virtual Server group. While a MapService is associated with an ImageServer or FeatureServer, it will automatically access the QueryServer, GeocodeServer, and ExtractServer when required. No extra administration is needed to take advantage of these three Virtual Servers.

When a MapService is requested, the Application Server delivers the request to a Spatial Server within a Virtual Server group. If an ArcIMS Spatial Server goes down, incoming requests can still be processed by other Spatial Servers assigned to the same Virtual Server. In turn, additional Spatial Servers can be added to a Virtual Server to help meet increased loads.

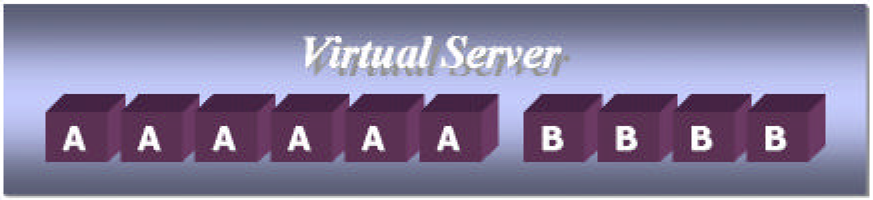
Example Showing Spatial and Virtual Servers

The diagram (below) shows two Spatial Servers, A and B, that have been grouped into the same Virtual Server. When a MapService is assigned to this Virtual Server, it is started on both Spatial Server A and Spatial Server B (Feature MapServices would be assigned to a FeatureServer and Image MapServices would be assigned to an ImageServer). When the Application Server receives a request, it looks at which Virtual Server the MapService has been assigned to. In this example, the request is forwarded to either Spatial Server A or B.



Virtual Servers are used for administration and include one or more Spatial Servers.

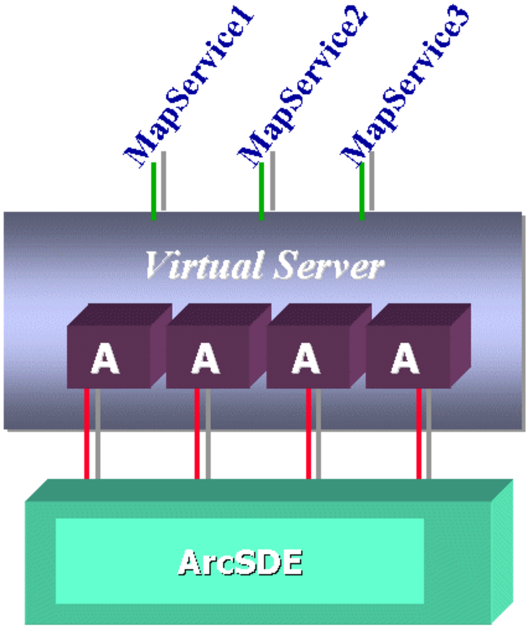
When a Spatial Server is assigned to a Virtual Server, all instances of that Spatial Server are included. If Spatial Server A has six instances and B has four instances, then the Virtual Server has access to ten total instances. In this scenario, an incoming request for a MapService can be sent to any of the ten instances running.



A Virtual Server is made up of multiple instances of a Spatial Server.

Relationship of Spatial Servers and ArcSDE

If ArcSDE is used as a data source, to work properly with ArcIMS one ArcSDE connection should be available for each instance of an ArcIMS Spatial Server. In the example below, Spatial Server A has four instances running. The number of ArcSDE connections needed is also four—one connection for each instance of Spatial Server A.



The number of ArcSDE connections equals the number of Spatial Server instances.

The number of MapServices using ArcSDE is independent of the number of ArcSDE connections needed. The three MapServices shown (above diagram) will use all four ArcSDE connections. If ten MapServices were running, they would use the same four ArcSDE connections.

Multiple MapServices can use the same group of ArcSDE connections if

- The same Virtual Server is used.
- The same ArcSDE instance is used.

Basically, the number of ArcSDE connections required is related to the number of Spatial Server instances running, not the number of MapServices running.

Connections to ArcSDE are also needed for the QueryServer, which is required in conjunction with ImageServers. (By default one QueryServer is needed for every ImageServer.) In the above example, if Spatial Server A is an ImageServer, four ArcSDE connections would be needed for the ImageServer and four connections would be needed for the QueryServer for a total of twelve connections. Additional ArcSDE connections are needed for the GeocodeServer and ExtractServer if called upon by a MapService.

ArcIMS is a trusted client to ArcSDE, which means that an unlimited number of connections with ArcSDE are available. However, if needed, ArcIMS does support connection pooling to reduce the number of connections with ArcSDE. When pooling is used, two or more instances of a Spatial Server can share one connection with ArcSDE.

ArcIMS Manager

The easy-to-use ArcIMS Manager is a suite of Web pages that provides access to all ArcIMS serverside functions and tools. Through the ArcIMS Manager, users can quickly set up and administer Internet services. Simple instructions guide users through the steps of authoring, designing, and publishing map services.

The ArcIMS Manager consists of three stand-alone components to

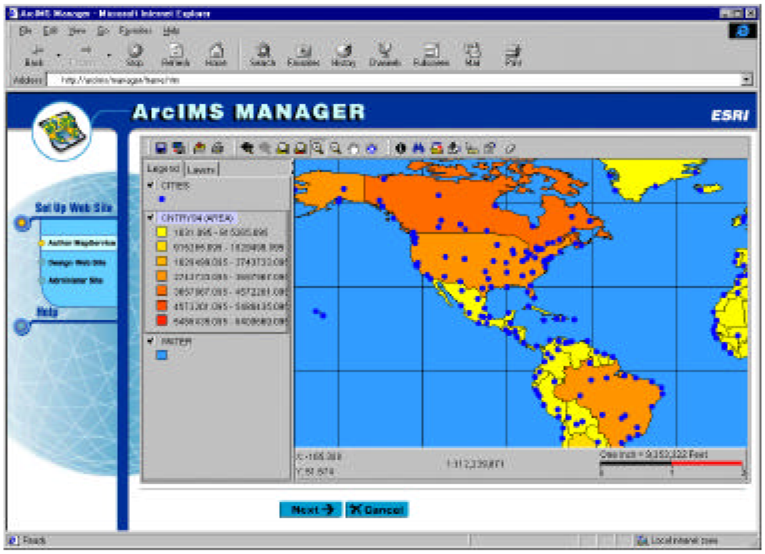
- Author MapService files.
- Design Web pages.
- Publish MapServices and administer ArcIMS Spatial Servers.

The ArcIMS Manager combines these applications—the ArcIMS Author, ArcIMS Designer, and ArcIMS Administrator—into one wizard-driven framework. Although each of these processes is available as an independent application that runs outside a Web browser, the ArcIMS Manager provides the transitional steps for additional communication among the components. The ArcIMS Manager can be used remotely to administer a site.

Authoring MapService Files

Maps can be created using the ArcIMS Author, a self-contained, menu-driven mapping applet used to define map characteristics. During this process, shapefiles, ArcSDE data sets, and images are loaded; connections to databases are made; symbology is defined; scale dependencies are set; and other mapping parameters are defined. The output from Author is a MapService configuration file written in ArcXML. An ArcXML file can also be edited in a text editor.

ArcIMS Author



Here is a sample of the MapService configuration file generated from the map image (shown above).

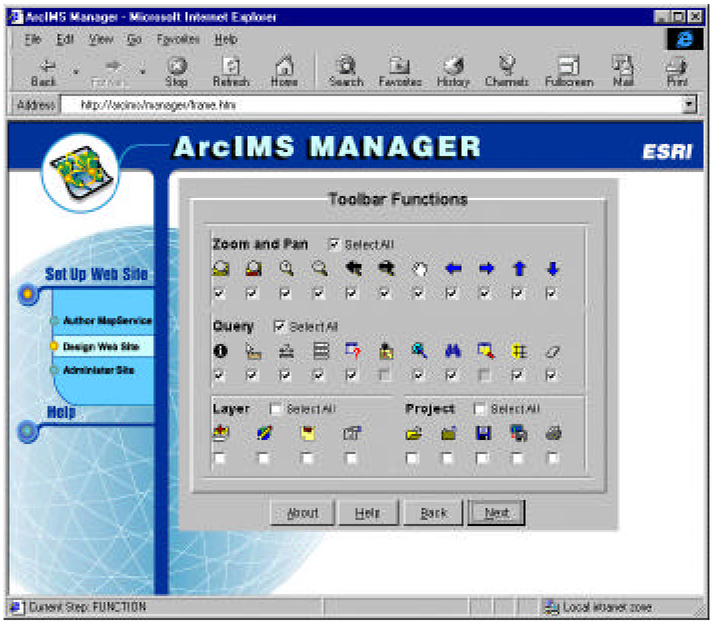
```
<?xml version="1.0" encoding="Cp1252"?>

<ARCXML version="1.0.1">
  <CONFIG>
    <MAP>
      <PROPERTIES>
<ENVELOPE minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"
name="Initial_Extent" />
<MAPUNITS units="DECIMAL_DEGREES" />
      </PROPERTIES>
      <WORKSPACES>
<SHAPEWORKSPACE name="shp_ws-0" directory="C:\data" />
      </WORKSPACES>
<LAYER type="featureclass" name="CNTRY94" visible="true" id="1">
<DATASET name="Countries" type="polygon" workspace="shp_ws-0" />
<SIMPLERENDERER>
  <SIMPLEPOLYGONSYMBOL filltype="solid"
fillcolor="255,255,153"/>
</SIMPLERENDERER>
      </LAYER>
...
    </MAP>
  </CONFIG>
</ARCXML>
```

Designing Web Pages

The Designer is used to generate Web pages that allow a user to interact with a map. The Designer leads the user through a series of panels for selecting which MapServices to use, which template style to use, and which operations and functions will be available in a client Web browser. Designer has three template options: an HTML Viewer, a customizable Java Viewer, and a noncustomizable Java Viewer. The Viewers are described in more detail in the ArcIMS Components: Client Viewer section.

ArcIMS Designer


ArcIMS Manager - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print
ArcIMS MANAGER
ESRI
Set Up Web Site
Author MapService
Design Web Site
Administer Site
Help
Toolbar Functions
About Help Back Next
Local intranet zone

Address http://arcims/manager/frame.htm

Zoom and Pan ☑ Select All

Query ☑ Select All

Layer ☐ Select All

Project ☐ Select All

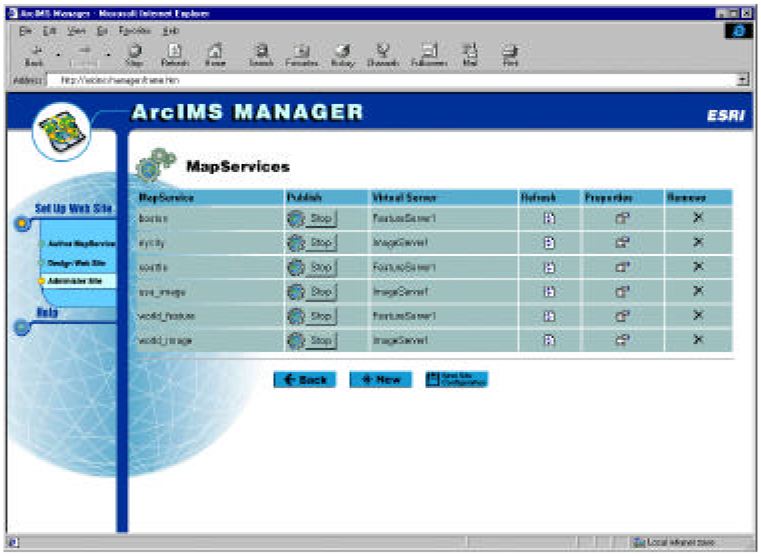
Current Step: FUNCTION

The output from Designer is a series of HTML pages. The Web pages can be used as-is, enhanced for a unique look or to meet specific needs.

Publishing MapServices and Administering ArcIMS Spatial Servers

The ArcIMS Manager has tools for administering the ArcIMS site for MapServices, Spatial Servers, and Virtual Servers.

ArcIMS Administrator



MapServices

MapService	Publish	Virtual Server	Refresh	Properties	Remove
borders	Stop	FeatureServer1			✕
mycity	Stop	ImageServer1			✕
seattle	Stop	FeatureServer1			✕
usa_image	Stop	ImageServer1			✕
world_feature	Stop	FeatureServer1			✕
world_image	Stop	ImageServer1			✕

MapServices

Before a MapService becomes available to client requests, it must be published. Through the Manager, MapServices can be added, started, stopped, refreshed, and removed.

Spatial Servers

In a typical install, ArcIMS starts with one Spatial Server running. Using the Administrator, additional Spatial Servers can be started on the same machine or different machines. New instances of existing Spatial Servers can also be started.

Virtual Servers

When ArcIMS is installed, five Virtual Servers are created: ImageServer, FeatureServer, QueryServer, GeocodeServer, and ExtractServer. Additional Virtual Servers can be added and existing ones can be deleted. Performance can be monitored by checking statistics such as response time.

ArcIMS is designed so that MapServices, Spatial Servers, and Virtual Servers can be added and removed while the ArcIMS site continues to operate. A site configuration can be saved so that when there is downtime, the site will automatically restart with the same configuration.

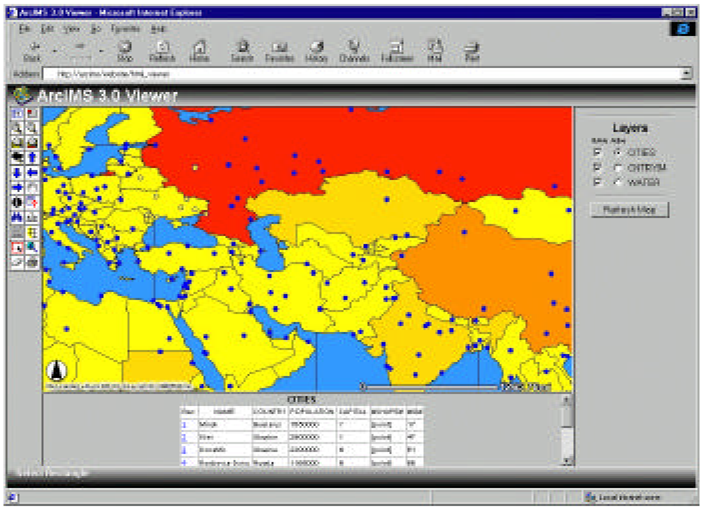
ArcIMS Components: Client Viewers

The ArcIMS Viewers can either be entirely HTML or include Java Applets. ArcIMS comes with a HTML/DHTML Viewer and two Java Viewers generated using the Designer and sample HTML Viewers using the ColdFusion and ActiveX Connectors. ArcIMS also comes with a stand-alone Java Viewer called ArcExplorer 3. The viewers can be modified using HTML and JavaScript and new client viewers.

HTML/DHTML Viewer

The HTML/DHTML Viewer is written using HTML, DHTML, and JavaScript. In this environment only one Image MapService can be displayed at a time. When a user clicks on a map or tool, a request is generated by the viewer and sent to an ArcIMS Spatial Server using the Servlet Connector. When a response is returned, the client parses the response for display instructions. The HTML/DHTML Viewer requires that browsers must be version 4.0 or higher to handle the communications for requests and responses. Support for lower version browsers is available by creating a customized viewer.

ArcIMS HTML/DHTML Viewer



Viewers Using the ColdFusion or ActiveX Connectors

ColdFusion and ActiveX Viewers are also HTML/DHTML implementations. These differ from the HTML Viewer in that all processing is handled on the serverside. Requests are made through third party application servers, and the response is an HTML page generated on the fly. The viewer does not need to generate a request or parse the response, making a thinner client. To the end user, these viewers may look identical and appear to have similar functionality, but the underlying handling of requests and responses is quite different. ArcIMS includes sample viewers for use with both the ColdFusion and ActiveX Connectors.

Java Viewers

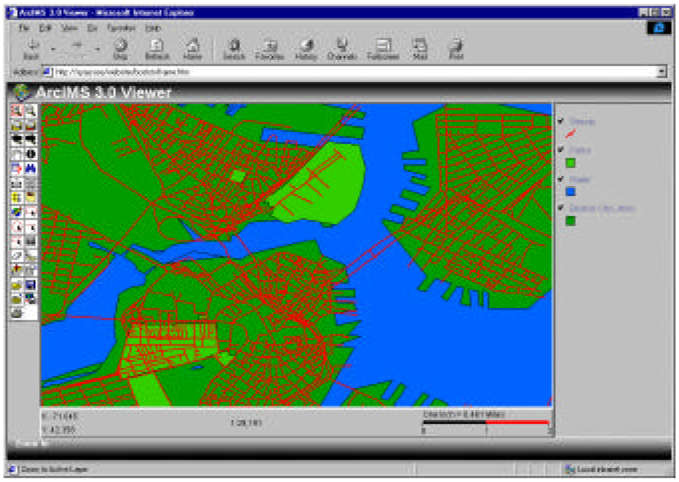
As compared to the HTML/DHTML Viewers, the Java Viewers are thicker clients because these support both Image and Feature MapServices. Multiple MapServices can be combined with local data and viewed in the same Java Viewer. The viewers use a Java 2 Applet for displaying the information and handling requests.

The Java Viewers support feature streaming and more clientside processing. Data that is streamed to the Java Viewers is temporarily cached on the client machine. Requests are handled on the client machine unless the request requires data that is not currently in the cache. In this instance, the request is sent to the server to either retrieve more data or process data residing on the server. The temporary cache is removed when the viewer is closed.

ArcIMS includes two Java Viewers: Java Custom and Java Standard. Both have similar functionality.

The Java Custom Viewer uses JavaScript to communicate with the applets. This viewer can be customized using methods in a Viewer Object Model API. Currently, the Java Custom Viewer is supported only in Internet Explorer 4.0 and 5.0.

ArcIMS Java Custom Viewer



The Java Standard Viewer does not use JavaScript. The tools and functions are predefined and cannot be customized using the Viewer Object Model. The Java Standard Viewer is supported on both Netscape and Internet Explorer 4.0 and higher browsers.

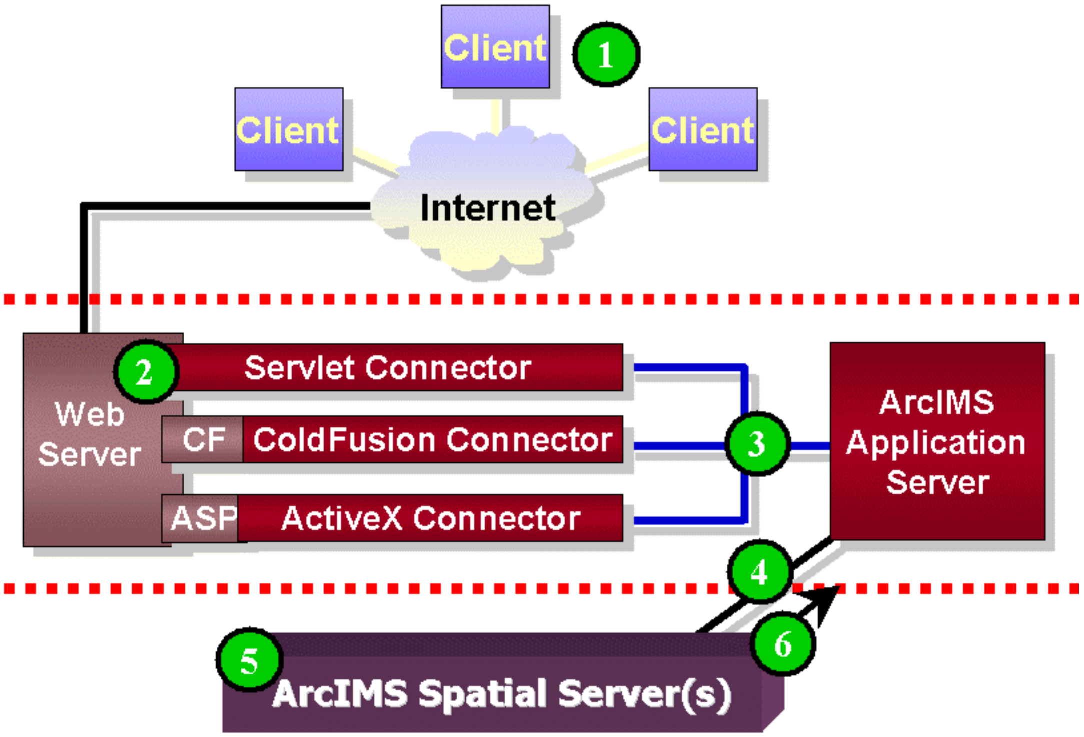
ArcIMS also ships with ArcExplorer 3, a stand-alone viewer that does not require a browser. Java Standard Viewer tools and functions are predefined and cannot be customized.

Java Viewers necessitate two downloads. The first is the Java Run-time Environment, which is required when Java 2 Applets are used. The second is a onetime download for the ArcIMS viewer components so that the applet can communicate with the server. The viewers can be customized to always download the components from an ArcIMS site in lieu of the onetime download.

How ArcIMS Communicates

When an ArcIMS client makes a request for a map or data, the request travels first to the Web server and then through the connectors and Application Server to a Spatial Server. The responses follow the same path in reverse order. All requests and responses are written using ArcXML.

Steps Taken by ArcIMS to Process a Request and Send a Response



Step	*Action*
1	Client sends a request to an ArcIMS site.
2	The Web Server receives the request and passes it to the Servlet Connector or ColdFusion/ASP servers, which in turn hand the request to a connector.
3	The Connector opens a path for the ArcIMS Application Server to respond, and the request is handed from the connector to the Application Server.
4	The Application Server sends the request to an available Spatial Server within a Virtual Server group.
5	The Spatial Server generates the response as a • Response XML string (such as query results or an image location) • Stream of data
6	Response returns through the reverse order of the initial request.

Summary

ArcIMS has a multitiered architecture consisting of

- Presentation tier—ArcIMS clients
- Business logic tier—Web Server, ArcIMS Application Connectors, and the ArcIMS Application Server
- Data storage tier—ArcIMS Spatial Server

These tiers communicate through ArcXML. In addition to handling requests and responses, ArcXML is used for MapService configuration files.

When a request is received by the Web Server, it is handed to one of three ArcIMS Application Connectors:

- Servlet
- ColdFusion
- ActiveX

ArcIMS requests that are processed through the Servlet Connector are immediately handed to the ArcIMS Application Server. The Application Server handles load distribution and keeps track of which MapServices are running on which ArcIMS Spatial Servers. When ColdFusion and ASP are used, the incoming page is handled by the ColdFusion and ASP application servers, respectively. When an ArcIMS request is processed, the request is sent to the Application Server through a connector.

The backbone of ArcIMS is the Spatial Server. The server provides five functions: image rendering, feature streaming, geocoding, querying, and feature extraction. Spatial Servers are not accessed directly but rather through Virtual Servers. Virtual Servers consist of one or more Spatial Servers for administration purposes. An incoming request for a MapService goes to one of the Spatial Servers within the Virtual Server group for which the MapService has been assigned.

ArcIMS offers Image and Feature MapServices. Image MapServices use the image rendering capabilities of a Spatial Server. Maps are generated on the server and sent to a client as a GIF, JPEG, or PNG image. Feature MapServices take advantage of the Spatial Server feature streaming functions. Data is bundled on the server and streamed across to a client.

ArcIMS supports HTML and Java clients. Java Viewers use a Java 2 Applet and support both Image and Feature MapServices. Java Viewers contain more clientside processing capabilities, support feature streaming, and support multiple MapServices and local data within the same viewer. HTML Viewers are lighter weight, but only one Image MapService can be viewed at a time. HTML Viewers can use the Servlet Connector and can also take advantage of the ColdFusion and ActiveX Connectors.

Users access the different components of ArcIMS using the Author, Designer, and Administration tools. These tools are accessible through the Manager but are also available as stand-alone components.

The ArcIMS architecture has been developed specifically for Internet applications. It is designed to handle small Intranet sites as well as the industrial scale needs of enterprisewide systems or e-business sites. ArcIMS can scale to meet server capacity needs as Web site demand increases. Additional Spatial Servers can be added quickly to existing Virtual Servers. ArcIMS is also designed to work easily with other Internet tools and applications.